TURBO news

Revista para Computadores ATARI Nº 8 - MARZO 1990

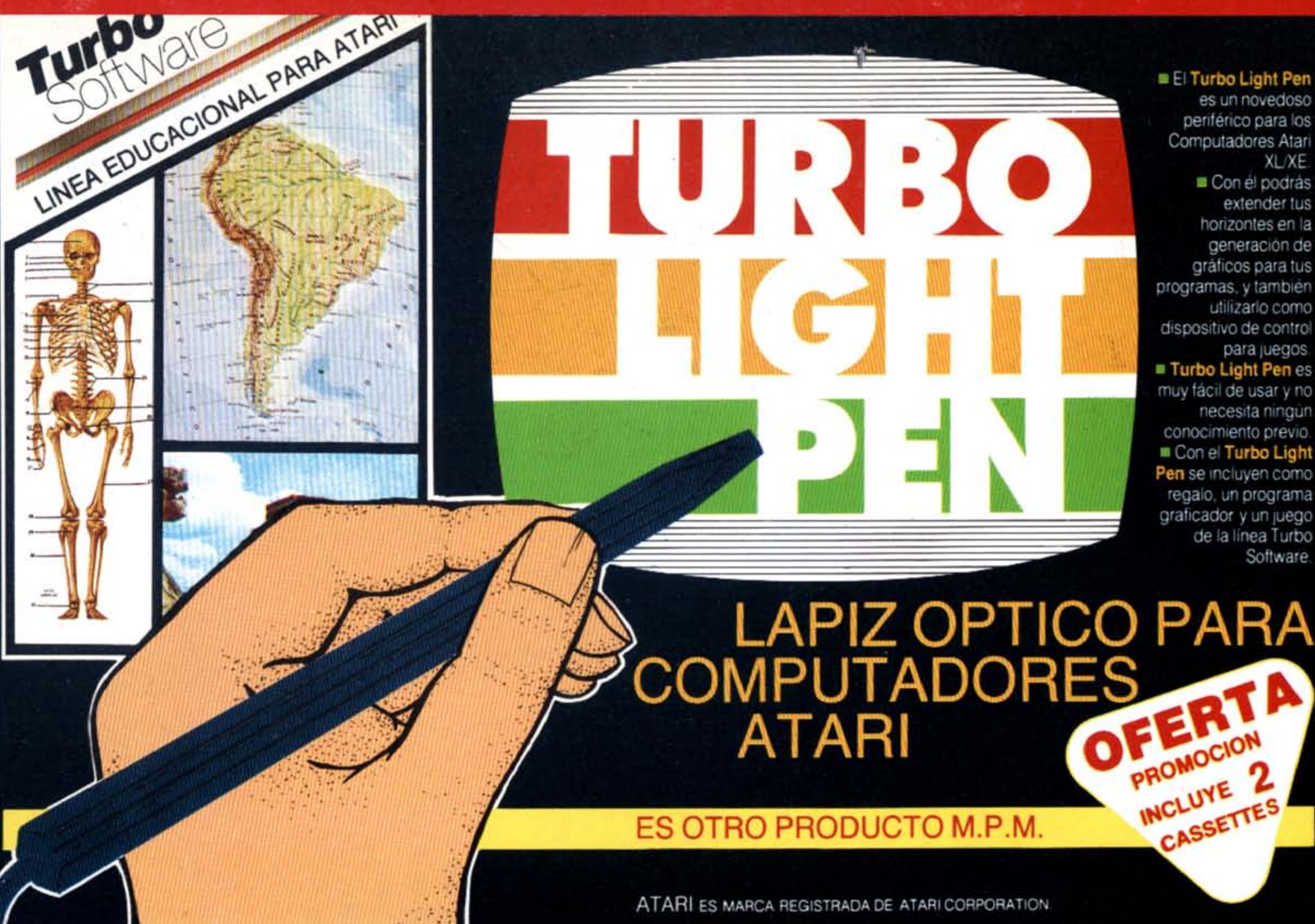
$ 550

ATARI

ATARI

ATARI

ATARI



ADQUIERELOS EN LOS SIGUIENTES PUNTOS DE VENTAS

• **ANTOFAGASTA:** COOPERCARAB KW VIDEO LA ESPAÑOLA • **VIÑA DEL MAR:** FALABELLA VIÑA INSIS MPR COMPUTACION • **VALPARAISO:** COMPUTRONIC • **SANTIAGO:** AUDIO BICICLETA INTERNAC CASA ROYAL CENTRO ATARI COMERCIAL ESTADO COMPUMANQUE COMPUCENTER FALABELLA AHUMADA FALABELLA P. ARAUCO IMACO INFOGROUP PC STORE PETERSEN ROLEC SUPERMERCADOS UNIMARC TASCO VIDEO CLUB INTERNACIONAL • **RANCAGUA:** CASA ZUÑIGA • **CURICO:** MULTIHOGAR • **TALCA:** LIBRERIA "EL AHORRO" MULTICENTRO VIDEO CLUB CASSAL • **CHILLAN:** CASA EDISON • **CONCEPCION:** COOPERCARAB DISMAR DISMAR 2 EQUS PHANTER RAPSODIA SESCO • **LOS ANGELES:** DISTRIBUIDORA MERINO • **ANGOL:** SCORPIO • **VICTORIA:** CASA SIGMUND • **TEMUCO:** COMERCIAL MANQUEHUE ESTABLECIMIENTOS GEJMAN FALABELLA • **PUCON:** ELTIT • **VILLARRICA:** JOYERIA KETTERER • **VALDIVIA:** ELECTROMUSICA • **LA UNION:** IMPORTADORA COSMOS • **OSORNO:** CASA REAL FOTO EXPRESS • **PUERTO VARAS:** ELECTRO HORN • **PUERTO MONTT:** COMERCIAL MANQUEHUE

EDITORIAL

Estimados amigos:

Las vacaciones ya han quedado atrás, por lo que enfrentamos el nuevo año totalmente renovados y con más fuerza que nunca. Esperamos que las hayan disfrutado intensamente y que hayan reunido muchas ganas para continuar juntos en la tarea de descubrir los secretos de nuestro gran amigo. Tenemos grandes proyectos para este año. Nuevos productos dedicados especialmente al Hobbista, se están preparando en estos mismos instantes y se anuncian para un futuro muy cercano. Estén atentos que en el próximo número ya les podremos dar más detalles de sus respectivos lanzamientos. El avance de la tecnología está llegando también a este área, y estamos seguros que va a ser un año sumamente fructífero para todos los amantes del ATARI.

A partir de este número hemos incorporado, ya en forma definitiva, el artículo correspondiente a los juegos de ingenio, donde podrás ir demostrando los conocimientos adquiridos a través de nuestras páginas.

Para los amantes del desarrollo de hardware, no se desesperen. Simplemente no podemos atender todos los requerimientos que nos llegan al mismo tiempo.

CONTENIDO

TURBO news ®

Circulación Mensual, Nacional e Internacional.
Destinada a los usuarios de computadores ATARI (R) como material didáctico de. Programación. TURBO news (R) es una publicación de EDITORA TURBO LTDA. Domicilio: Av. Fco. Bilbao 4226 - Teléfono: 486506.

DIRECTOR RESPONSABLE: Mauro Pieressa. REPRESENTANTE LEGAL: Marcelo Waldbaum. PRODUCCION: Marcelo Waldbaum y Mauro Pieressa, Programadores y Diseñadores de Computación. DIRECTORA DE ARTE: Odali Guerrero L. CORRECTOR: Marcial Valenzuela S. PUBLICIDAD Y RR.PP.: Liliana Muñoz Otárola, Hernán Vittini. COLABORACION: Mariana Pizarro. PUZZLE: Mario Calvo A. FOTOCOMPOSICION: Brubytes. IMPRESION: Servigraf. DISTRIBUCION: Alfa Ltda. Agradecemos la colaboración de COELSA S.A. Centro Atari. (Augusto Leguía Sur 75). Profesor Emilio Antileff. Atari es marca registrada de ATARI CORPORATION. TURBO news es marca registrada de EDITORA TURBO LTDA. (Reg. Marca N° 342428 9-05-89).

TURBO MAIL

Sres. Turbo News:

Aprovechando su espacio "Turbo Mail", les escribo con el objeto de manifestarles algunas dudas que se me han presentado en mis inicios en la computación:

1. Escribí un programa para el cálculo de curvas hidrostáticas y luego de correrlo, lo grabé en un cassette. La dificultad que no he podido resolver es que al intentar cargarlo me arroja el error Nro. 143 en la línea 0 y no puedo recuperarlo.

2. Quisiera que me expliquen la forma de crear mis propios archivos de datos en mi cassettera.

Mi computador es un 800-XL y la cassettera es la XC-12.

Mario Sandoval Sáez
Río Juncal 62
Denavi Sur
Talcahuano

Estimado Mario:

El problema de la grabación de programas Basic en cassettes es muy común, pero utilizando Cintas de muy buena calidad puedes solucionar este inconveniente. Además es aconsejable grabar más de una vez el programa como respaldo de la información evitando este tipo de problemas.

Lamentablemente es imposible recuperar el programa que generó un error en la carga, así que tendrás que reingresarlo nuevamente en el Computador.

En la Edición Nro. 3 de TURBO news, describimos la utilización de los comandos Basic "CSAVE y CLOAD", así que les recomendamos revisar el artículo del Curso de Basic para analizar si utiliza bien la sintaxis de estas instrucciones.

En cuanto a la generación de archivos, tanto en cassettes como en diskettes, pensamos incorporar un curso especial para tratar este tema dentro de las lecciones del Lenguaje Basic, pues es uno de los conceptos más importantes del Computador y en la información de los manuales que acompañan al 800 XL está bien confuso.

Sres. TURBO news:

Aprovecho la oportunidad para felicitarlos por su excelente revista que me permite enfocar mi computador desde otro punto de vista: el desarrollo de Software y Hardware.

Me interesaron mucho las lecciones de Assembler, Gráficos por computadora y los desarrollos de Hardware, los cuales encontré muy interesantes.

Mi consulta específica es si existe la posibilidad de desensamblar todo el Sistema Operativo para obtener un listado detallado en Assembler del S.O. para así poder analizarlo conjuntamente con su curso de Lenguaje Assembler. En dicho caso ¿cómo puedo hacer para listarlo y qué herramientas necesito?

Pedro Martínez

Estimado Pedro:

Gracias por darnos aliento para seguir adelante en este proyecto de comunicación directa con la comunidad de atarianos.

Por suerte existen varias herramientas de Software y Hardware que nos permiten listar el Sistema Operativo del Atari. En versión de disco las más utilizadas son el Diskwizar y el Disassembler y en cartucho, los lenguajes Assembler como el MAC/65 y el Editor de Assembler, cuentan con esta posibilidad.

Pero el problema que tienen estos programas radica en que sólo podemos obtener con ellos listados y no nos permiten interactuar con los programas que estamos desensamblando.

Existe un sistema combinado de Software y Hardware que tiene la posibilidad de ocupar y listar al Sistema Operativo. Es la herramienta más apta para introducirse dentro del S.O. Lo encontramos en el mercado con el nombre "Omnimon" y consiste en una memoria ROM que se conecta reemplazando o no al Sistema Operativo del Computador dependiendo de un interruptor que lo habilita y lo deshabilita.

Debido a la importancia que posee el Omnimon dentro de las herramientas que todo programador Atari necesita, incluimos en esta Edición de TURBO news, un artículo que lo describe detalladamente para que nuestros lectores puedan ver el alcance que este sistema nos otorga.

MANTENTE EN CONTACTO



LECCION 8

En este número veremos cómo podemos crear archivos en disco o cassette, para evitar perder toda la información cuando apagamos el computador.



En números anteriores, hemos visto cómo se podían guardar los programas que realizábamos, de manera de poder recuperarlos para su posterior utilización. Pero hasta ahora, no habíamos visto cómo guardar los datos que híbamos cargando. Los almacenábamos en un vector o matriz, sobre los que podíamos cargar y consultarlos, pero al apagar el programa o el computador, debíamos recomenzar nuevamente todo el proceso.

Antes de comenzar a ver las instrucciones en Lenguaje Basic, describiremos las dos formas de archivo que existen en ATARI.

Los archivos pueden ser secuenciales o directos. En el primer caso los datos se almacenan uno a continuación del siguiente. O sea que para llegar al cuarto dato, por ejemplo, debemos primero haber leído los tres anteriores. En cassette, es la única alternativa que existe, ya que el grabador no puede posicionarse



automáticamente en algún lugar del archivo. En disco, debido a su facilidad de ubicarse en cualquier punto de éste, (ver artículo sobre diskettes del número anterior) esto puede obviarse y es por ello que tenemos el modo directo. En él uno puede acceder directamente al dato seleccionado, sin haber pasado por los anteriores.

Primero veremos el modo secuencial.

Lo primero que debe hacerse es comunicarle al computador nuestro deseo de trabajar con archivos, ya sea en disco o cassette. Durante ese proceso debe indicársele cómo se llama el archivo (solamente en el caso de usar diskette), si en él se va a sólo leer, sólo grabar, leer y grabar datos, etc. y el número de canal que se va a utilizar en la comunicación. Atari posee ocho canales de comunicación, numerados del 0 al 7, de los cuales el 0 lo utiliza para escribir en pantalla, el 6 para escribir en pantalla usando gráficos y el 7 para las instrucciones LPRINT, CLOAD y CSAVE. Es decir que del 1 al 5 quedan disponibles para el programador.

Todo lo anteriormente mencionado se hace con la instrucción OPEN como se ve a continuación:

INSTRUCCION OPEN

Se escribe de la siguiente manera:

OPEN #nro de canal, operación, auxiliar, archivo.

El número de canal, como ya vimos, puede variar de 1 a 5

Las operaciones pueden ser las siguientes:

4 Operación de entrada.
8 Operación de salida.
12 Operación de entrada/salida.
6 Búsqueda del directorio del disco.
9 Operación de grabar datos a continuación del último que haya hido grabado.

El código auxiliar no se utiliza casi nunca. Depende del periférico que se esté utilizando. En La impresora Atari 820, por ejemplo, un código 83 significa que debe imprimir lateralmente.

En el campo archivo se debe poner entre comillas sobre qué dispositivo deseamos trabajar y en el caso de ser disco, cuál es el nombre del archivo.

Los dispositivos posibles son:

"P:" IMPRESORA
"C:" CASSETTERA
"D:n nombre.extensión"
DISKETTERA n Y ARCHIVO nombre.extensión
"K:" TECLADO
"E:" EDITOR DE PANTALLA
"S:" MONITOR DE PANTALLA

Estos tres últimos explicaremos más adelante para qué sirven.
Como ejemplo de OPEN veremos los siguientes ejemplos:

OPEN #1,4,0,"D:NOMBRES.1"

Prepara el archivo NOMBRES.1 para LEER datos a través del canal 1.

OPEN #3,8,0,"D:NOMBRES.1"

Prepara el archivo NOMBRES.1 para grabar en él, datos a través del canal 3. Cabe destacar que si este archivo ya existía, éste se pierde totalmente. Por eso la existencia del código de operación 9, que permite que se conserve el archivo anterior y que lo que se va a grabar se haga a continuación.

OPEN #2,4,0,"C:"

Prepara el grabador para leer datos desde el canal #2.

OPEN #5,8,0,"P:"

Prepara la impresora para imprimir. Cabe destacar que la impresora sólo acepta el código de operación 8. No tendría sentido ponerle otro código.



BASIC

OPEN #3,6,0,"D:*.*"

Se utiliza para poder leer el directorio del disco, como veremos en el ejemplo que acompaña el artículo.

Una vez que se le ha indicado al computador qué operación se va a realizar y por cuál dispositivo, comienza el verdadero proceso de lectura o grabación de datos.

Ello se logra con las siguientes instrucciones:

INSTRUCCION PRINT #nro canal, variable

Esta instrucción se utiliza para enviar información a través del canal dado. Como el computador sabe, debido a la instrucción OPEN, para qué sirve ese canal, enviará la información al dispositivo correspondiente. Dicha información es la que se encuentra contenida en la variable. Al igual que la instrucción PRINT común, pueden haber varias variables en una sola instrucción PRINT #nro. así como constantes puestas entre comillas.

Veamos unos ejemplos:

```
PRINT #1,NOMBRE$;TELEFONO$
PRINT #1,NOMBRE$,TELEFONO$
```

La diferencia entre estos dos casos está en la cantidad de espacios en blanco que el computador dejará entre ambas variables.

```
PRINT #1,"EL NOMBRE
ES";NOMBRE$,"Y SU
EDAD";EDAD
```

La instrucción utilizada para leer datos es la siguiente:

INSTRUCCION INPUT #nro. canal, variable.

Esta instrucción es similar a la instrucción INPUT normal, sólo que se le especifica un canal para indicarle desde dónde van a tomar los datos.
Ejemplo:

```
INPUT #1,NOMBRE$
INPUT #4,DATO
```

Es importante destacar cómo funciona internamente todo este proceso. El computador, al grabar, junta todas las variables en una sola (todo esto el programador no lo ve) y forma con ella un registro que es el que envía al dispositivo correspondiente. Esto tiene importancia en la posterior lectura de ese registro. Cuando se lee, se debe leer esa única variable (cuya longitud es igual a la suma de los tamaños de todas las variables y tomando la longitud de las numéricas como 13), en una auxiliar y luego asignarle a cada una de las variables el pedazo que le





corresponda de dicha variable auxiliar. Si alguna de las variables es numérica, en la grabación se graba como numérica, pero en la lectura se debe tomar el VAL del pedazo correspondiente.

Por ejemplo si hacemos:

```
PRINT #1;NOMBRE$;DOMICILIO$;
TELEFONO
```

Y suponiendo que NOMBRE$ y DOMICILIO$ tienen una longitud de 20 y teléfono una de 13 (por ser numérica), deberíamos hacer cuando queramos leer el archivo:

```
INPUT #1,AUX$
```

Siendo AUX$ una variable de longitud 53. A continuación debemos hacer:

```
NOMBRE$=AUX$(1,20):DOMICILI
O$=AUX$(21,40):TELEFONO=VAL
(AUX$(41))
```

Otra forma de hacer este mismo proceso, sería generar un registro por cada una de las variables, como se ve a continuación:

```
PRINT #1;NOMBRE$
PRINT #1;DOMICILIO$
PRINT #1;TELEFONO
```

Y para leer:

```
INPUT #1;NOMBRE$
INPUT #1;DOMICILIO$
INPUT #1;TELEFONO
```

Notar que en este último caso, no hizo falta utilizar la instrucción VAL, ya que el registro estaba ocupado por una sola variable numérica y la





transformación fue automática.

Para indicarle al computador que dejamos de usar un archivo se debe usar la instrucción CLOSE.

INSTRUCCION CLOSE #nro. canal

Esta instrucción se utiliza para indicarle al computador que puede desocupar ese canal y utilizarlo para alguna otra cosa. Es imprescindible hacer un CLOSE al concluir la utilización de algún archivo sino, se puede prestar para errores.

Ejemplo:

```
CLOSE #1
CLOSE #3
```

Si un archivo fue abierto con una instrucción OPEN para lectura y deseamos luego que sirva para grabación, debemos primero hacer un CLOSE para luego hacer el OPEN correspondiente a la nueva operación.

Cuando terminamos de grabar un archivo simplemente hacemos un CLOSE y ya está. Pero cuando estamos leyendo un archivo, evidentemente no vamos a saber qué tan largo va a ser. Lo que se hace es decirle que continúe leyendo mientras pueda. Cuando se termine y marque un error, debe haber una instrucción TRAP que lo prevenga y que haga que se ejecute una instrucción CLOSE.

Daremos, como primer ejemplo de aplicación, un programa que permite leer un directorio del disco desde BASIC. Para ello utilizamos el código de operación número 6. El programa sería el siguiente:

```
10 DIM A$(20)
20 OPEN #1,6,0,"D:*.*"
3f TRAP 50
40 INPUT #1,A$:PRINT A$:
GOTO 40
50 CLOSE #1
```

Al ejecutar este programa con la instrucción RUN, se desplegarán en pantalla todos los archivos que contiene cada disco. A$ tiene una longitud de 20, ya que cada nombre de archivo más su extensión y su tamaño ocupan 20 caracteres. Como se ve, aquí se queda leyendo títulos indefinidamente hasta que marca error, el cual fue controlado con la instrucción TRAP, por ello sigue en la 50 donde cierra el archivo.

En la parte de programas, al final de la revista, veremos el mismo programa visto en números anteriores para crear una agenda telefónica, pero con la diferencia que esta vez guardará todos sus datos en un archivo de disco. Si posees cassettera en todos los lugares donde dice "D:AGENDA", deberás poner "C:". También tiene la opción de utilizar la impresora, si la tienes. Fíjate que el grueso del programa es exactamente igual al anterior, lo único que se hace es incorporarle los ciclos de almacenamiento, lectura e impresión de datos. Al final del programa aparecen reproducidas las rutinas de lectura y grabación, utilizando el segundo sistema de almacenamiento de información.En el próximo número veremos el otro sistema de almacenamiento de datos, es decir, el modo directo. ¡¡Nos vemos!!

PROMOCION *turbo*

APRENDIENDO A DIBUJAR CON LOGO

En este número veremos cómo poder realizar otras figuras geométricas además del cuadrado y del rectángulo y haremos algunos ingeniosos dibujos con ellos.

Antes de ver nuevas figuras veremos algunos sencillos gráficos que podemos realizar con nuestra vieja y conocida instrucción cuadrado. Recordemos primero cómo se construía esta instrucción:

```
CR CUADRADO :LADO
REPITE 4 [AV :LADO DE 90]
FIN
```

Primero haremos una estrella de cuadrados con la siguiente instrucción:

```
REPITE 8 [CUADRADO DE 45]
```

Con ésta simple instrucción veremos cómo se forma este complejo dibujo:

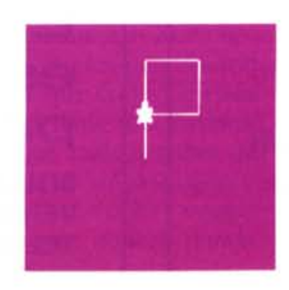


LECCION 6

Ahora veremos cómo hacer una bandera. Lo vamos a hacer de manera de generar una instrucción que se llame bandera para poder utilizarla más adelante:

```
CR BANDERA :LADO1
AV :LADO1
CUADRADO :LADO1
FIN
```

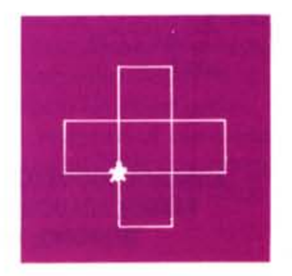
Cuando escribamos BANDERA 30 veremos lo siguiente:



Con la instrucción BANDERA se puede hacer el siguiente gráfico:

```
CR CRUZ :LADO2
REPITE 4 [BANDERA :LADO2
DE 90]
FIN
```

Cuando hagamos Cruz 30 va a quedar:

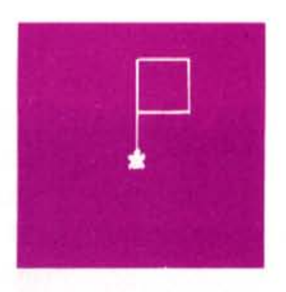




Vamos a hacer una variante de la instrucción BANDERA que nos va a ser útil para hacer las próximas figuras. Es básicamente parecida a Bandera pero la tortuga en lugar de terminar en el medio de la bandera va a terminar al principio de ésta. Veamos cómo queda:

```
CR BANDERA2 :LADO3
BANDERA :LADO3
RE :LADO3
FIN
```

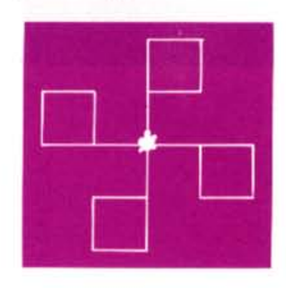
Si escribimos BANDERA2 50 va a quedar:



Con BANDERA2 podríamos hacer el siguiente dibujo:

```
CR BANDERAS :LADO4
REPITE 4 [BANDERA2 :LADO4
DE 90]
FIN
```

Al poner BANDERAS se dibujará lo siguiente:



Utilizando BANDERAS también podemos hacer otra variante:

```
CR MUCHASBANDERAS :LADO5
BANDERAS :LADO5
DE 45
BANDERAS :LADO5
FIN
```

Siendo el resultado el siguiente:

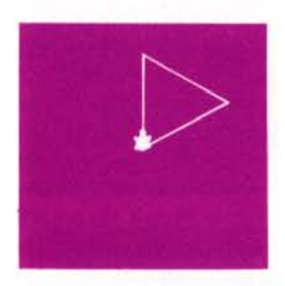


A continuación vamos a ver la siguiente figura geométrica que es el triángulo.

Este se construye de la siguiente manera:

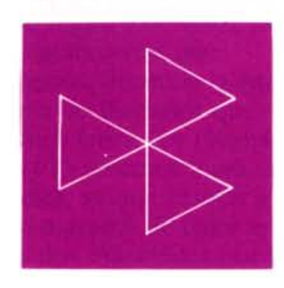
```
CR TRIANGULO :LADO
REPITE 3 [AV :LADO DE 120]
FIN
```

Cuando escribamos TRIANGULO 30 veremos lo siguiente:

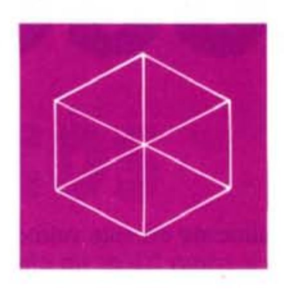


Con esta instrucción podemos hacer los siguientes gráficos:

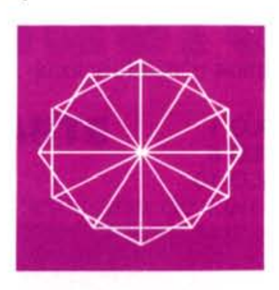
```
REPITE 3 [TRIANGULO 30
DE 120]
```



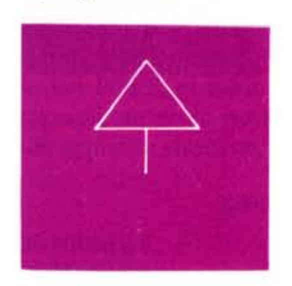
```
REPITE 6 [TRIANGULO 30 DE 60)
```

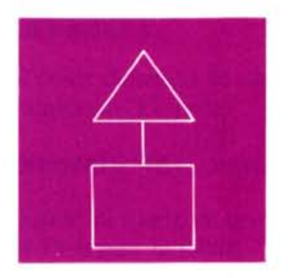


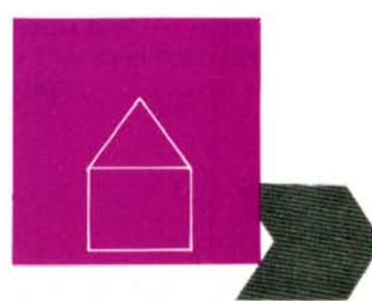
```
REPITE 12 [TRIANGULO 30
DE 30]
```



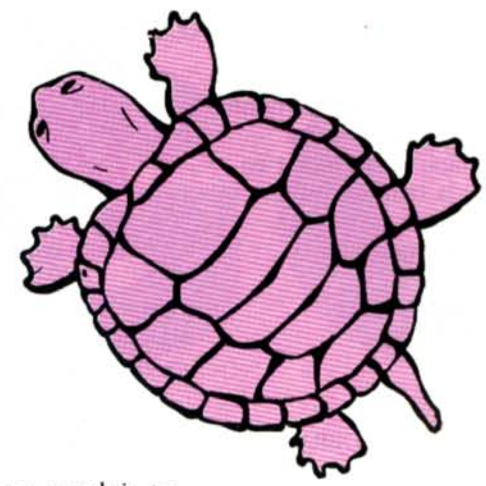
Trata de (utilizando la instrucción triángulo) realizar los siguientes gráficos:







LOGO



Finalmente en este número veremos cómo hacer un círculo.

```
CR CIRCULO :PASOS
REPITE 36 [AV :PASOS DE 10]
FIN
```

Veamos cómo funciona:

```
CIRCULO 1
CIRCULO 5
CIRCULO 10
```



Finalmente, para concluir, te dejamos los siguientes gráficos para que intentes realizarlos por tu cuenta.









Gráficos por computadora

TERCERA PARTE

Continuando con el curso de Gráficos por Computadora, veremos en este número cómo es el manejo del color, anchura, prioridad y colisiones de los Player-Missiles.

Cada uno de los Player-Missiles, tienen características que pueden modificarse en forma independiente. Estas son el movimiento, visto en el número anterior, el color, la anchura, la prioridad y la colisión. La prioridad es la que determina si el Player debe pasar por "encima" o por "debajo" de los dibujos de fondo u otros Players. Dichos dibujos son los que forman los escenarios donde se desenvuelven los juegos. En Atari el escenario está dividido en "campos de juego" o Playfields, cada uno de ellos formado por todas las figuras que son comandadas por el mismo registro de color. Del modo gráfico con que trabajemos, depende la cantidad máxima de Playfields que podemos tener. Por ejemplo, en gráfico 15 tenemos hasta cuatro Playfields, ya que existen sólo cuatro registros de color en este modo (los manejados por los registros 708, 709, 710 y 712). El registro de color 1, por ejemplo, controla el Playfield 1.

La colisión es lo que permite saber si algún Player está chocando, ya sea contra otro Player, algún Missile o bien algún Playfield. A continuación veremos cada uno de ellos.

Selección de color

El color de cada uno de los Player-Missiles puede manejarse en forma independiente con el uso de un registro de memoria por cada uno de ellos. Para modificar cualquier color basta con asignar, mediante la instrucción POKE, el valor correspondiente en la posición deseada según la siguiente tabla.

	DIRECCION
PLAYER-MISSILE 0	704
PLAYER-MISSILE 1	705
PLAYER-MISSILE 2	706
PLAYER-MISSILE 3	707

El color deseado se obtiene de la siguiente fórmula:

COLOR=MATIZ*16+LUMINANCIA

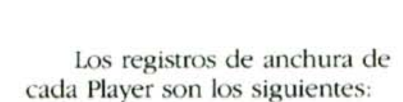
Donde el matiz es un valor de 0 a 15 según la tabla desplegada más abajo y luminancia también es un valor de 0 a 15 y da el brillo del color respectivo.



Gráficos por computadora



NRO. MATIZ	COLOR
0	GRIS
1	AMARILLO
2	NARANJA
3	ROJO ANARANJADO
4	ROSADO
5	PURPURA
6	PURPURA AZULADO
7	AZUL OSCURO
8	AZUL CIELO
9	AZUL CLARO
10	TURQUESA
11	VERDE AZULADO
12	VERDE
13	VERDE AMARILLO
14	NARANJA VERDOSO
15	NARANJA CLARO



Observe que cada Player con su correspondiente Missile comparten el mismo color.

Para obtener un Player-Missile multicolor, puede hacerse de dos maneras. La primera de ellas, y la más efectiva, es mediante interrupciones de Display-List, descrita en números anteriores pero, para utilizarla, hay que trabajar en lenguaje Assembler. La segunda y más sencilla es superponer dos Players para formar un único dibujo. Por ejemplo, dibujar con un player un auto de carreras de color azul y con otro las ruedas, de color negro. Cuando debamos mover el auto tendremos que mover simultáneamente los dos players para que parezca uno solo. Se puede obtener un tercer color de esta combinación como veremos un poco más adelante.

Selección de anchura del Player

Las posibilidades en cuanto a anchura de los Players son las siguientes: Simple, Doble o Cuádruple, según veremos en los siguientes dibujos.

Los registros de anchura de cada Player son los siguientes:

	DIRECCION
PLAYER 0	53256
PLAYER 1	53257
PLAYER 2	53258
PLAYER 3	53259
MISSILES	53260

Los valores que se deben Pokear en éstas son los siguientes:

Para anchura simple: 0 ó 2
Para anchura doble:1
Para anchura cuádruple: 3

Cuando uno prende el computador, estas posiciones toman el valor 0, lo mismo sucede si se oprime la tecla RESET.

Control de Prioridad

Como ya dijimos, la prioridad es utilizada para indicarle al computador, si un Player determinado debe pasar por encima o por debajo de los demás Players y de los campos de juego. Con esto se pueden hacer excelentes efectos visuales, como por ejemplo cuando una persona entra a una casa y desaparece, por el simple hecho de tener una prioridad más baja que la pared y que pueda asomarse tranquilamente por la ventana sin más esfuerzo que poner cuidado en las prioridades. En realidad no desapareció, sino que quedó "debajo" del dibujo con mayor prioridad. Todo esto es manejado por la posición de memoria 623, que tiene cuatro valores posibles, según la siguiente tabla:

VALOR	PRIORIDAD
1	P0 P1 P2 P3 TODOS LOS PLAYFIELDS
2	P0 P1 TODOS LOS PLAYFIELDS P2 P3
3	TODOS LOS PLAYFIELDS P0 P1 P2 P3
8	PF0 PF1 P0 P1 P2 P3 PF2 PF3



Donde Pn significa Player número n y PFn, Playfield número n. Lamentablemente no se pueden realizar todas las combinaciones de Players con Playfields, sin embargo poniendo especial cuidado en cómo se escogen la numeración de los players y de los Playfields se puede hacer cualquier cosa con estas cuatro combinaciones.

La posición de memoria 623 puede utilizarse también para lograr un tercer color en la combinación de Players, tema que había quedado pendiente en la explicación de los colores vistos más arriba. Si le sumamos 32 al valor de la prioridad escogida, cuando se superponen dos players en lugar de uno ocultar al otro, se unen para formar un tercer color.

Detección de Colisiones

Cuando un auto choca contra una pared, cuando una bala hiere a un Sheriff o cuando un misil destruye una nave enemiga, necesitamos enterarnos al instante. Es por esto que Atari posee 16 registros de colisiones que se actualizan 16 veces por segundo que nos indican si algún Player o algún Missile está chocando contra algo. Los 16 registros son los siguientes:

FUNCION	DIRECCION
PLAYER 0 CON PLAYER	53260
PLAYER 1 CON PLAYFIELD	53253
PLAYER 1 CON PLAYER	53261
PLAYER 2 CON PLAYFIELD	53254
PLAYER 2 CON PLAYER	53262
PLAYER 3 CON PLAYFIELD	53255
PLAYER 3 CON PLAYER	53263
MISSILE 0 CON PLAYFIELD	53248
MISSILE 0 CON PLAYER	53256
MISSILE 1 CON PLAYFIELD	53249
MISSILE 1 CON PLAYER	53257
MISSILE 2 CON PLAYFIELD	53250
MISSILE 2 CON PLAYER	53258
MISSILE 3 CON PLAYFIELD	53251
MISSILE 3 CON PLAYER	53259

Para determinar si hubo alguna colisión bastará con examinar, con la instrucción PEEK, el valor de estas posiciones de memoria. Pueden darse los siguientes valores:

Si colisionó con Player o Playfield 0=1
Si colisionó con Player o Playfield 1=2
Si colisionó con Player o Playfield 2=4
Si colisionó con Player o Playfield 3=8

Por ejemplo si la posición 53251 da 2, quiere decir que el Missile 3 chocó con el Playfield 2. Si la 53260 da 8 significa que el Player 0 chocó con el Player 3.

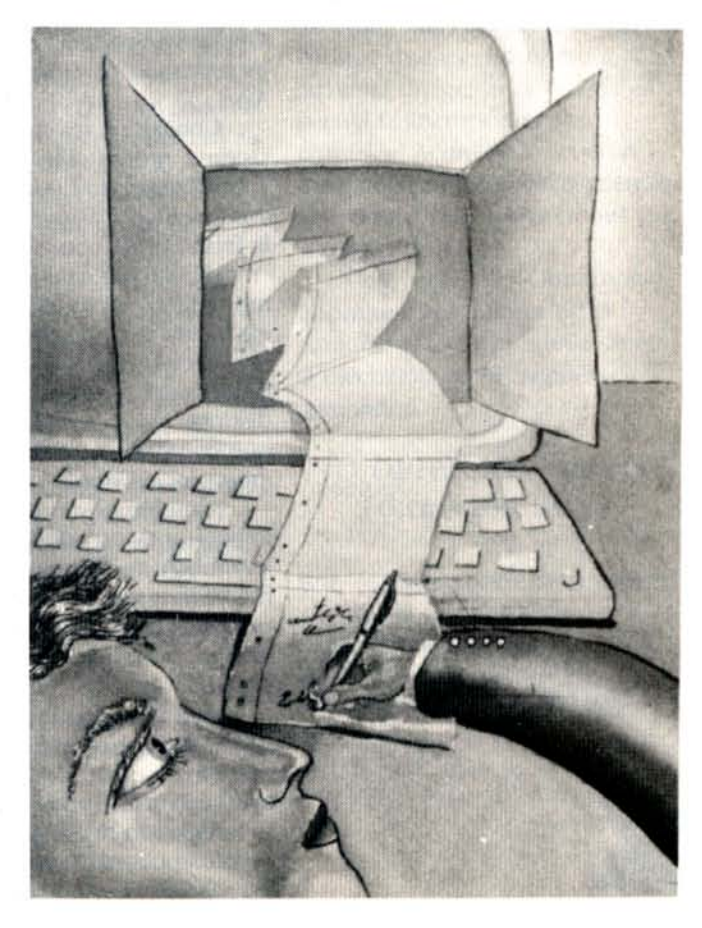
También pueden darse combinaciones. Por ejemplo si quiere preguntarse si el Player 1 chocó con el Playfield 1 y 3 al mismo tiempo, hay que sumar los valores correspondientes a estos dos campos. Habría que preguntar si la posición 53253 es igual a 10 (2+8).

Las posiciones 53248 a 53245 son nuestras viejas y conocidas posiciones de memoria, utilizadas para dar la posición horizontal de los Player-Missiles. Vemos que si Pokeamos sobre ellas modificamos la posición horizontal pero si las leemos a través de Peek nos da los registros de colisiones. Esto provoca que si queremos borrar un registro de colisión, poniéndolo en cero, debemos hacerlo de otra manera. Esto es pokeando el valor 0 en la posición de memoria 53278.

Cuando hacemos:
POKE 53278,0.

Todos los registros de memoria se borran automáticamente.

En el próximo número ya estaremos listos para realizar los primeros ejemplos en el uso de Player-Missiles.



OMNI

una herramient
para el pr

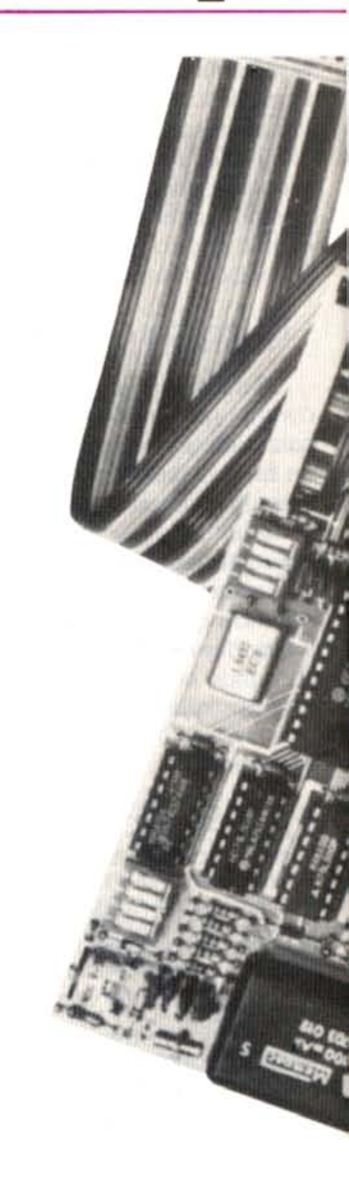
Hemos recibido muchas consultas en nuestra Editorial sobre los sistemas que nos permiten analizar y tomar el control absoluto del computador para obtener información que vaya más allá de la que se puede obtener en los manuales del computador. Debido a esta situación presentaremos en este artículo el mejor sistema que hasta el momento se haya desarrollado para interactuar con el 6502: "el OMNIMON".

El OMNIMON es un Sistema Operativo ampliado que una vez instalado en el computador, está siempre disponible para el usuario reemplazando, cuando éste lo desee, al Sistema Operativo Standard que viene incorporado de fábrica en el Atari.

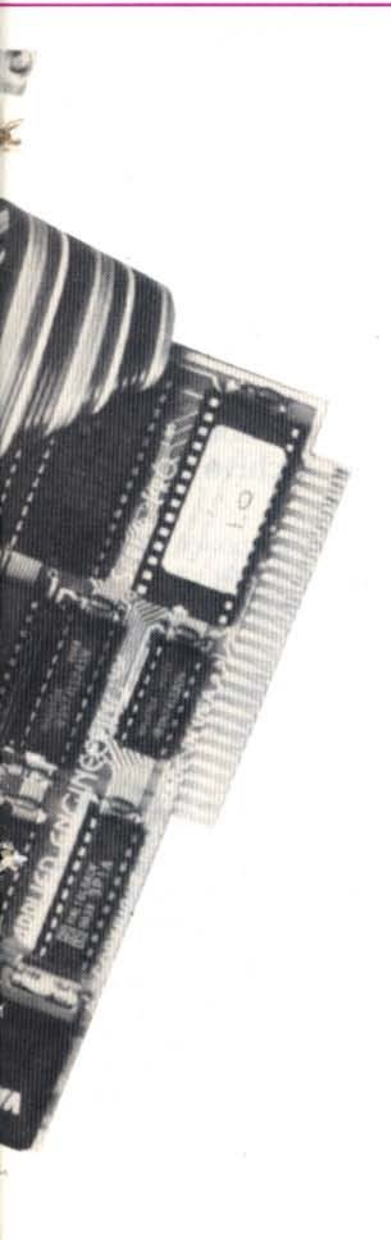
Una vez funcionando este sistema, otorga al usuario el control absoluto de su computador. Este control incluye la facilidad de consultar y "modificar" en cualquier momento y en medio de la ejecución de cualquier programa el contenido de los Registros del procesador como así también las distintas posiciones de memoria que conforman el mapa de memoria del Atari.

También permite realizar impresiones de listados en Assembler de cualquier zona de memoria y ejecutar subrutinas ubicadas en la memoria para analizar su funcionamiento paso a paso.

Si te has propuesto la necesidad de aprender Assembler, el Omnimon te ayudará muchísimo en la investigación de programas ya escritos por programadores avanzados, además siempre te



MON:
a indispensable
gramador.



permite analizar la ejecución de tus programas en caso de producirse errores en el listado.

Otra de las aplicaciones importantes del Omnimon es la posibilidad que otorga este sistema en el análisis y modificación de programas almacenados en disco. Con él la tarea de convertir programas que carguen con la estructura de file a boot es muy sencilla, así como también el trabajo inverso. Generalmente estas conversiones son aplicadas cuando queremos que algún archivo cargue directamente de un cassette teniendo el programa original en disco.

Cuando en el computador tenemos instalado el Omnimon, en realidad éste opera normalmente hasta que el usuario desee que el control del Atari pase temporalmente al Omnimon para luego retornar al programa en curso. Esto se logra presionando las teclas SELECT y RESET. Al hacerlo, en la pantalla aparece la presentación del Omnimon, y se despliega el contenido de los registros de la CPU:

```
PC   NV-BDIZC AC X  Y  SP
AF24D  73     00 09 3F 1FF
```

PC: Indica el contenido del Program Counter, es decir la próxima instrucción que la CPU debe ejecutar.

NV-BDIZC: Indican los valores de los flags del Procesador.

AC: Contenido del acumulador.

X: Contenido del registro X.

Y: Contenido del registro Y.

SP: Este valor contiene el Stack Pointer, que es el puntero a la próxima dirección de la pila en la cual se incorporará o se retirará el Acumulador con las instrucciones PHA o PLA además del resto de las instrucciones que afectan a la Pila.

Una vez que el Omnimon imprime estos valores en la pantalla, el usuario puede elegir la ejecución de los siguientes comandos:

D (dirección inicial) (dirección final):

Este comando permite al usuario analizar el contenido de determinada zona de memoria. Si queremos ver la zona comprendida entre $200 y $224



OMNIMON



debemos ingresar el comando:

D 200 224

A (dirección) byte byte byte. . . :

Con este comando podemos modificar cualquier posición dentro de la memoria del computador. Lo único que debemos hacer, es ingresar la dirección inicial y los bytes que deseamos incorporar a partir de dicha posición.

S (dirección inicial) byte byte. . . :

A veces es muy útil ubicar dentro de un programa una sucesión de bytes o texto para modificarlos, sobre todo cuando el programa ha sido desarrollado por otro programador. Para esto es muy útil el comando S, que nos indica en qué dirección se encuentra la sucesión de bytes solicitados a partir de una dirección de origen. Por ejemplo, en el Sistema Operativo del computador, si queremos ubicar el primer LDA #$00 debemos ingresarle al Omnimon el siguiente comando:

S C000 A9 00

Si le está indicando al Omnimon que el comando deseado es el de búsqueda, C000 es la dirección de inicio del Sistema Operativo y A9 00 es la instrucción LDA #$00 compilada. Luego de ingresarle este comando, nos contesta con todas las apariciones de esta instrucción dentro del Sistema Operativo.

R (sector) (dirección de carga) (cantidad de sectores):

Con esta instrucción, el Omnimon lee directamente de la Unidad de Discos sin necesidad que el usuario haya cargado el DOS. Para ocupar esta instrucción, debemos indicarle al Omnimon el sector inicial de lectura, la dirección en la cual va a dejar los bytes leídos y la cantidad de sectores en hexadecimal, tomando en cuenta que cada sector tiene 128 bytes.

W (sector) (dirección de grabación) (cantidad de sectores):

Esta instrucción se ocupa para grabar cualquier área de memoria en el disco. Al igual que la instrucción de lectura, se le indica el sector inicial de grabación, la zona de memoria de donde se toman los bytes a grabar y la cantidad de sectores a grabar.

G D: nombre del archivo.

Con esta instrucción, el usuario puede cargar en memoria un archivo en lenguaje de máquina.

J dirección de la rutina:

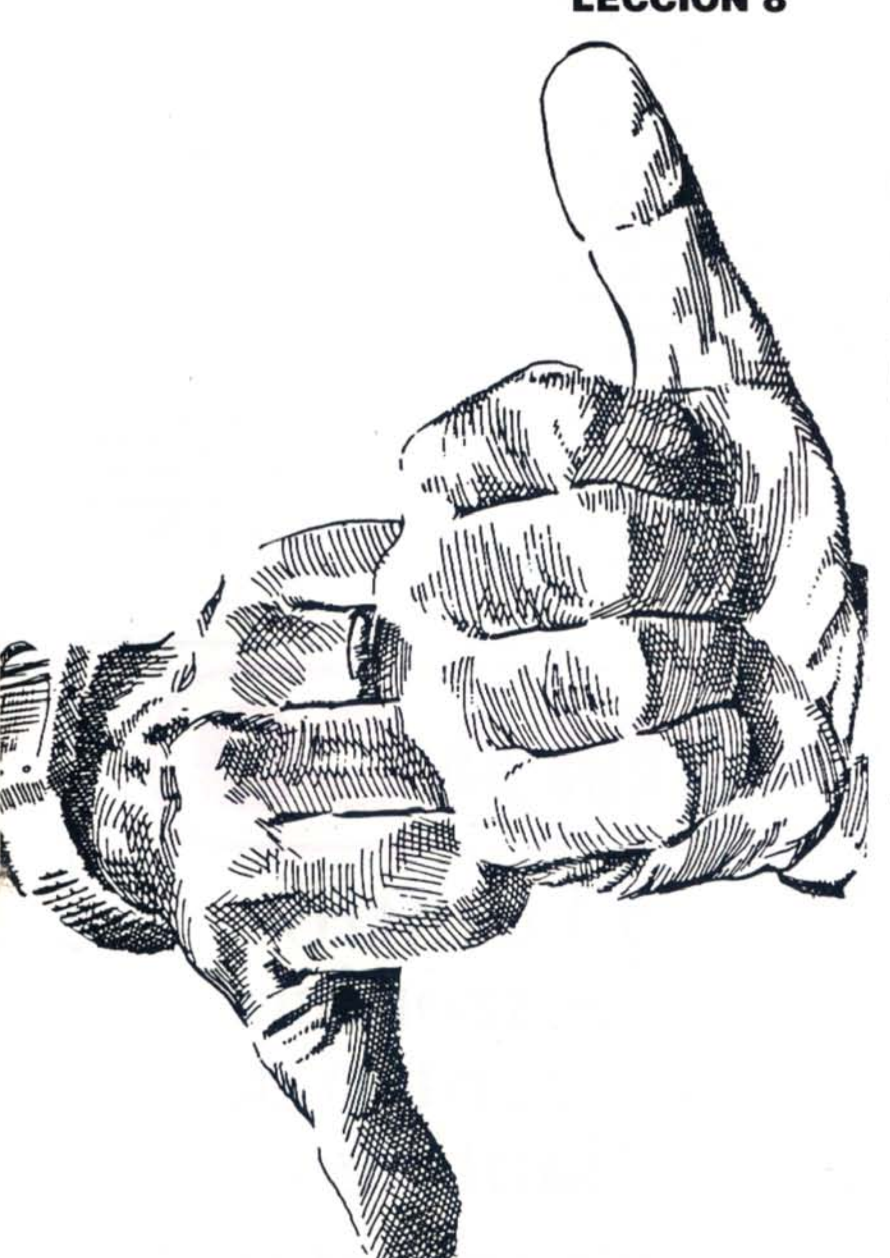
Este es uno de los comandos más importantes del Omnimon. Con él podemos ejecutar cualquier dirección dentro del mapa de memoria y retornar, luego de hacerlo, al Omnimon. Por ejemplo, podemos ejecutar el comando J $F2B0, lo cual se traducirá en un JSR $F2B0.

La intención de este artículo es presentar el Omnimon en su aspecto técnico y nuestra idea es seguir desarrollando en los números siguientes las aplicaciones prácticas que este producto nos presta a los programadores Assembler...

ASSEMBLER

ASSEMBLER

LECCION 8



En la edición número 7 de nuestra Revista TURBO news, iniciamos el desarrollo de un sistema de programación totalmente desarrollado en Lenguaje Assembler. Explicamos su estructura general, con su diagrama de flujo y en esta lección y las siguientes estudiaremos en detalle las distintas estructuras que componen el Programa.

ASSEMBLER

Si aceptaste el reto de seguir analizando el desarrollo del lenguaje para el cartel LIXXON DMP-32, tendrás en estos momentos muchas dudas. Nuestra idea es ir analizando cada una de las rutinas que en él están incorporadas, haciendo referencia a las subrutinas del Sistema Operativo que se utilizan en cada una de ellas.

En esta edición también incorporamos en la sección de programas de la revista, el listado de la subrutina que maneja el almacenamiento de las instrucciones en la memoria manteniéndolas ordenadas ascendentemente por el número de la instrucción.

Comenzaremos definiendo la estructura del lenguaje y las posibilidades que le brinda al usuario para que puedas entender el por qué de algunas de las rutinas que componen el programa.

Inicialmente, al cargar el programa, el computador despliega el mensaje que contiene la presentación del programa. Luego se queda en un ciclo esperando que el usuario ingrese una instrucción o bien un comando. Como ejemplo incorporaremos esta pequeña secuencia:

```
100 VELOCIDAD10
110 CON LA REVISTA
120 VELOCIDAD8
130 TURBO news, PODRAS
AVANZAR
140 EN EL APRENDIZAJE DE TU
150 COMPUTADOR ATARI
160 SET2
170 TURBO NEWS-TURBO
NEWS
180 SET1
```

El lenguaje que el cartel acepta, esta constituido por caracteres a desplegar a través del conjunto de leds y comandos especiales de gráficos. El sistema los diferencia, pues los comandos están totalmente escritos en video inverso. Los mensajes a desplegar por el cartel son escritos directamente en las líneas del programa. Así, una línea debe comenzar con su número de instrucción seguida por un mensaje en video normal, el cual será transmitido directamente al cartel, o bien un comando en video inverso, que primero será analizado para ver si está correcto.

En el programa principal del sistema, "EDITOR.MAC" que se encuentra listado en la edición 7 de tu revista, en la página 29, en la línea 4360 comienza una tabla de definición de los comandos válidos para el cartel. Esta tabla está definida de tal forma que puedan agregarse más comandos en el cartel sin tener que modificar la estructura del sistema. Los comandos que el cartel acepta son:

Pausa:

Este comando es utilizado para que el cartel se detenga unos instantes al encontrar este comando y el tiempo de pausa esta definido en la misma instrucción agregándole un valor que lo define. Para definir esta instrucción podemos ingresar la instrucción completa o bien su abreviatura:

100 PAUSA5
100 P5; abreviatura
100 PAUSA15
100 P15; abreviatura

Destello:

Este comando hace que los mensajes siguientes hasta encontrar el comando fin-destello pase en el cartel destellando a una velocidad determinada por el número de destello.

100 DESTELLO 8
110 TURBO NEWS
120 FIN DESTELLO

Con este programa el cartel desplegaría el mensaje "TURBO NEWS", destellando a la velocidad 8.

Gráfico:

Permite definir un caracter especial en la pantalla del cartel dándole la configuración de bits con números hexadecimales.

Velocidad:

Define la velocidad con la cual se despliegan los mensaje: en el cartel.

Set:

El cartel puede trabajar con dos Sets de caracteres, el set normal y el extendido, los cuales se seleccionan con los comando SET1 y SET2.

Borrar:

Este comando borra automáticamente la pantalla del cartel, cualquiera sea su contenido. Su abreviatura es "BO".

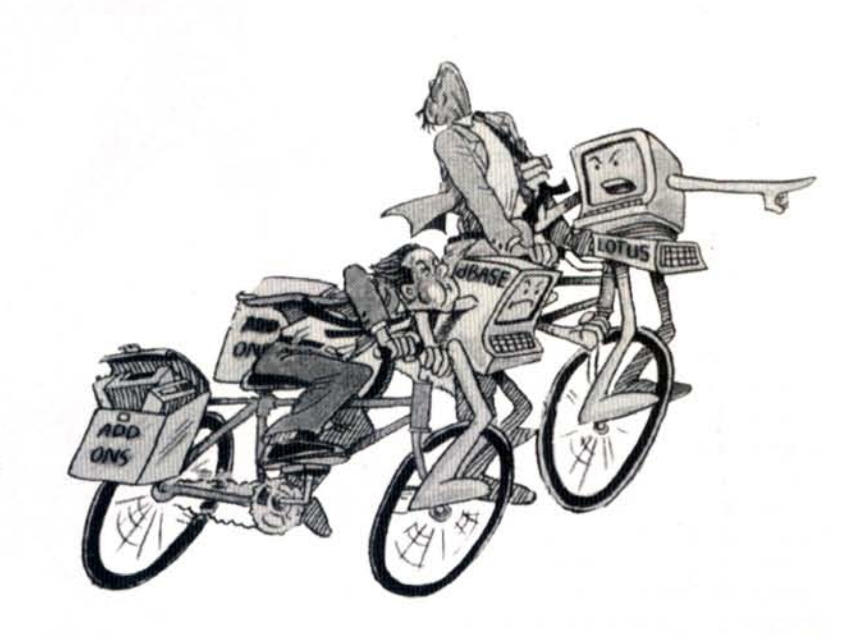
Con imagen - sin imagen:

Estos dos comandos hacen que la imagen se desconecte y conecte cualquiera que sea su contenido.

En el ejemplo 1, al recibir la primera línea del programa, el lenguaje detecta que se trata de una instrucción al cartel y no un comando hacia el computador, pues comienza con un número de instrucción. Al ser procesada y almacenada en la tabla de instrucciones el editor nuevamente espera otra línea del usuario. Al terminar de escribir el programa para el cartel, el computador está listo para seguir ingresando líneas o ejecutar algún comando. Si ingresamos el comando LIST, el computador listará en la pantalla el programa completo para el cartel. Si le definimos al computador el mensaje ENVIAR, éste le transmite al cartel todo el programa y, al recibirlo, el cartel comienza a desplegar los mensajes con sus respectivos comandos especiales de gráficos.

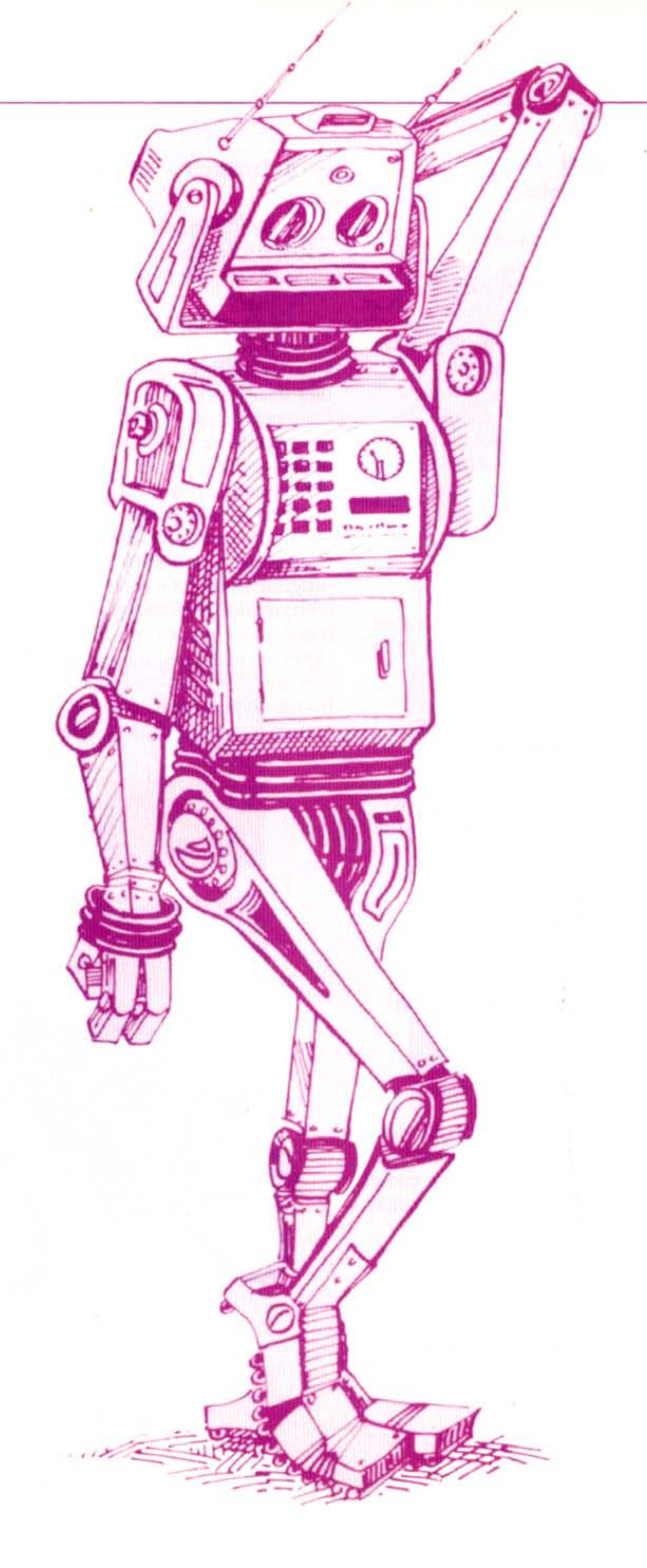
Una vez conocida la estructura general del lenguaje del cartel, comencemos a analizar en detalle el programa "EDITOR.MAC".

En las líneas comprendidas entre las instrucciones 180 y 640 se encuentran definidas las variables y posiciones de memoria que se ocuparán luego en el programa. Por ejemplo, en la línea 230 defino a POINTERAUX como la posición $A2. Además también reservo la posición $A3, no definiendo en esta posición ninguna variable. De esta forma, si quiero utilizar a





ASSEMBLER



la posición $A2 como un puntero auxiliar, es más cómodo recordarla por su nombre que por su dirección $A2.

Las líneas 680 y 690 de la página 25, definen la dirección del comienzo del programa. Cuando el computador Atari termina de cargar un programa, analiza el contenido de las posiciones de memoria $2E0 y $2E1 y salta a la dirección apuntada por estas posiciones. En este caso la línea 680 con el *= $2E0 está definiendo que las próximas instrucciones se compilen a partir de la posición $2E0. La instrucción 690 .WORD COMIENZO, está definiendo dos bytes que por la instrucción anterior se ubicarán en las posiciones $2E0 y $2E1. Estos bytes surgen de descomponer la dirección del rótulo COMIENZO que se encuentra en la instrucción 770 en su representación de HI y LO byte. Este trabajo lo hace automáticamente el Assembler al compilar el programa, él calcula la dirección del rótulo y lo define en el momento de cargarse el programa en las posiciones $2E0 y $2E1.

Luego de definir en el programa el nuevo set de caracteres y establecer algunas direcciones para el DOS, el programa llega al rótulo INICIO, en el cual se define a la variable FLAGNUM en cero representando esta situación que el sistema no se encuentra numerando automáticamente. Luego en las líneas siguientes a la 1390 se define la presentación del programa. Para esto, imprimimos

en la instrucción 1400 el caracter que se obtiene presionando las teclas ESC CONTROL CLEAR, que al imprimirlo en la pantalla, en realidad la borra por completo. La instrucción JSR $F2B0, es un llamado a una rutina del Sistema Operativo que imprime en la pantalla, en la posición en donde se encuentra el cursor, el caracter que se encuentra cargado en el acumulador.

Una vez borrada la pantalla, se imprime el contenido del mensaje ROTW que se encuentra definido en la línea 6530 de la página 31. Para este programa definimos la rutina imprimo que recibe un puntero MANDO y MANDO+1 que indica cuál es la primera posición de memoria que hay que imprimir en la pantalla. Esta rutina parte desde esta dirección e imprime los caracteres subsiguientes hasta encontrar un caracter *. Al encontrarlo para la impresión y devuelve el control al programa principal.

```
180 IMPRIMO
190     LDY #0
200     LOOPIMPRIMO
210     LDA (MANDO),Y
220     CMP #'*
230     BEQ FINIMPRIMO
240     STY GUARDOY
250     JSR $F2B0
260     LDY GUARDOY
270     INY
280     JMP LOOPIMPRIMO
290 FINIMPRIMO
300     RTS
```

Esta rutina es muy sencilla, primero define al registro Y en cero y realiza un ciclo leyendo la posición apuntada por la variable MANDO, que originariamente apunta al texto que deseamos imprimir. Una vez leída esta posición, la comparamos con el * para saber si debemos dejar la rutina y volver al programa principal. Si no es un *, lo imprimimos con la rutina del S.O. $F2B0, cuidando al registro Y, pues esta rutina pierde por su programación los valores de los registros, luego de imprimir el caracter en curso, vuelve a recuperar el registro Y, lo incrementa para pasar al próximo caracter y continúa en el ciclo de impresión. Es necesario para utilizar esta rutina tener mucho cuidado de incluir el caracter * en el texto, pues al no existir el computador imprimiría 255 caracteres que se encuentren luego de la posición apuntada por MANDO indefinidamente produciéndose un efecto inesperado.

Para definir el contenido de MANDO utilizamos las instrucciones comprendidas entre las líneas 1410 y 1440 con las cuales le dejamos al compilador hacer los cálculos necesarios acerca de las direcciones reales de memoria. LDA #<ROTW carga en el acumulador el LO BYTE de la posición ROTW y este byte lo almacenamos en MANDO y el LDA #>ROTW trabaja con el HI BYTE de ROTW. Luego lo único que nos queda es ejecutar un llamado a la Subrutina con un JSR IMPRIMO.

En esta lección continuamos analizando la estructura general del sistema y comenzamos a analizar las rutinas que lo componen. Puedes ya, con este material, comenzar a definir tus propios programas y en las siguientes ediciones de nuestra Revista seguiremos analizando detalladamente este sistema así como lo seguiremos listando para que ya lo puedas seguir completando.

TURBO ESTADISTICAS

Ranking Año 1989

A continuación damos el Ranking de Juegos de 1989. Este es el resultado de la unión de todos los rankings que hemos venido entregando a lo largo del año. Para el cálculo del puntaje se han otorgado 25 puntos por cada primer puesto, 24 por el segundo y así sucesivamente hasta llegar a uno por haber ocupado el vigésimoquinto lugar. El resultado es el siguiente:

PUESTO	TITULO	PUNTAJE
1	NINJA	138
2	MONTEZUMA'S II	134
3	INTERNATIONAL KARATE	126
4	SCREEMINGS WINGS 1942	113
5	POLE POSITION	93
6	LEADER BOARD GOLF	92
7	BRUCE LEE	91
8	GREAT AMERICAN RACE	87
9	RIVER RAID	86
10	BMX SIMULATOR	77
11	FLAK	64
12	FUTBOL II	61
13	MONTEZUMA	59
14	ELEKTRAGLIDE	53
15	STAR RAIDERS II	50
16	BLUE MAX	50
17	SWAT	44
18	NINJA MASTER	42
19	GHOSTBUSTERS	41
20	HENRY'S HOUSE	40
21	GHOSTCHASER	33
22	FIGHTER PILOT II	32
23	FUTBOL	30
24	BOULDER DASH II	29
25	MIRAX FORCE	29
26	OILS WELLS	28
27	ALLEY CAT	23
28	TANK COMMANDER	20
29	HARDBALL	20
30	LASER HAWK	20
31	ARKANOID	17
32	ZORRO	15
33	RESCUE ON FRACTALUS	14
34	DIMENSION X	14
35	POPEYE	13
36	PACMAN	11
37	KIKSTART	11
38	BOINA VERDE	10
39	WHO DARES WINS II	10
40	POOYAN	7
41	MARIO BROSS	7
42	RALLY SPEEDWAY	4
43	FROGGER II	3
44	RAID OVER MOSCOW	3
45	ZAXXON	3
46	POLAR PIERRE	2
47	CRYSTAL RAIDERS	1

Como recordarán, cada uno de los rankings que vamos entregando mes a mes es el resultado de la venta de cassettes Turbo Software en todo Chile sumado a sus preferencias. Sin embargo, existe un Ranking 1989 que puede ser formado con los puntajes otorgados por nuestros cronistas, a los tres juegos seleccionados y explicados en cada número. Dicho resultado es el siguiente:

PUESTO	CALIFICACION	TITULO
1	LEADER BOARD GOLF	6.875
2	HENRY'S HOUSE	6.825
3	MONTEZUMA'S REVENGE	6.775
4	NINJA	6.775
5	GHOSTCHASER	6.775
6	INTERNATIONAL KARATE	6.725
7	BMX SIMULATOR	6.7
8	GREAT AMERICAN RACE	6.675
9	FIGHTER PILOT II	6.675
10	STAR RAIDERS II	6.675
11	BLUE MAX	6.65
12	LASER HAWK	6.65
13	FLAK	6.625
14	SCREEMINGS WINGS 1942	6.625
15	POLE POSITION	6.525
16	RIVER RAID	6.475
17	FUTBOL II	6.35
18	BRUCE LEE	6.25



RANKING DEL MES

POS. DEL MES	POS. MES ANTERIOR	TITULO
1	9	SUPER SOCCER
2	2	MONTEZUMA'S REVENGE
3	5	NINJA
4	-	HARDBALL
5	4	GHOSTCHASER
6	-	CRYSTAL RAIDER
7	-	LEADER BOARD GOLF
8	17	TABLE SOCCER
9	7	INTERNATIONAL KARATE
10	3	THE EXTIRPATOR
11	1	HENRY'S HOUSE
12	-	RAID OVER MOSCOW
13	25	GREAT AMERICAN RACE
14	-	FIGHTER PILOT II
15	10	RIVER RAID
16	19	LITTLE DEVIL
17	5	TARZAN
18	-	BOULDER DASH I
19	-	ZAXXON
20	-	AMAUROTE
21	14	SWAT
22	12	SCREEMINGS WINGS 1942
23	24	FLAK
24	15	POLE POSITION
25	13	TWILIGHT WORLD

ASCENSO CONSTANTE DESCENSO

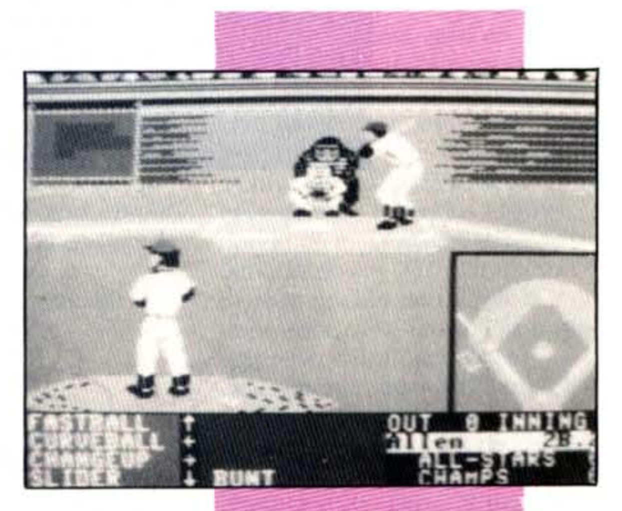
Este es el Ranking correspondiente a los meses de enero y febrero, obtenido en base a las estadísticas de ventas de cassettes Turbo Software en todo Chile. Recuerda que tus preferencias también serán tenidas en cuenta, para lo cual podrás escribir a Holanda 2456, Providencia, con los juegos de tu elección.

HARDBALL

Un juego para los amantes del Baseball, para ser jugado de a una o dos personas. El juego posee muchísimas variantes que dependerán de si eres tú el que lanza o el que batea la pelota. Dichas variantes se encuentran siempre descritas en la parte inferior de la pantalla y son ejecutadas moviendo el joystick en algunas de las ocho direcciones posibles.

En el primer caso, podrás elegir primero qué tipo de pelota deseas lanzar: curva, rápida, lenta, etc. Una vez elegido esto, sin que tu adversario lo haya notado, deberás elegir el lugar donde lanzarás la pelota. Si ésta es bateada, podrás elegir cualquiera de tus jugadores para que la atrapen y anulen así al adversario. El equipo bateador deberá oprimir el botón del joystick para indicar que está listo y luego mover el joystick en la dirección correspondiente para golpear la pelota. Si logras hacerlo, deberás rápidamente seleccionar el jugador que deseas que pase a la base siguiente.

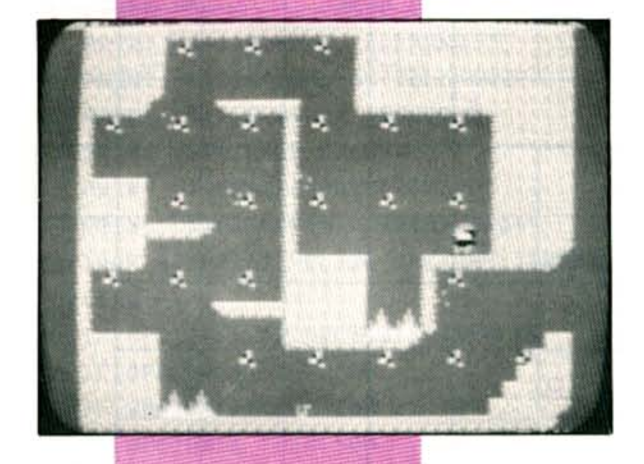
Sin dudas es un excelente simulador que, una vez adquirida la práctica necesaria, se vuelve sumamente atractivo.



SONIDO	6.5
GRAFICACION	6.7
ADICCION	6.7
PRESENTACION	6.7
PROMEDIO	6.65

CRYSTAL RAIDER

SONIDO	6.4
GRAFICACION	6.6
ADICCION	6.7
PRESENTACION	6.67
PROMEDIO	6.575



En este juego deberás conducir a nuestro pequeño explorador, a través de un intrincado laberinto recolectando las fabulosas joyas que allí se encuentran.

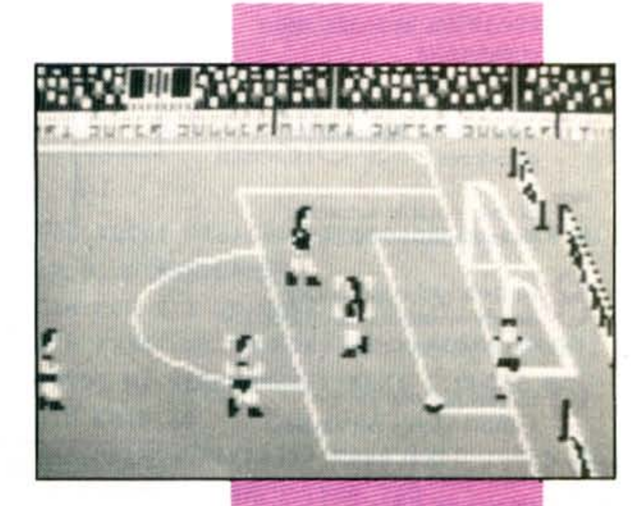
Para recogerlas deberás simplemente pasar por encima de ellas. Pero no todo es tan sencillo, los pasillos están plagados de peligros, como por ejemplo grandes fogatas, peligrosas serpientes, puentes movedizos, extrañas criaturas que patrullan el lugar, profundos precipicios etc.

Para poder eludirlos, deberás realizar grandes saltos oprimiendo el botón de tu joystick. Deberás tener en cuenta el tiempo límite que posees para terminar de recorrer cada una de las cuevas, ya que una vez vencido éste, morirás por falta de oxígeno. Puedes escoger el recorrerlas de día o de noche. En este último caso, poseerás una linterna que iluminará solamente tu alrededor.

SUPER SOCCER

Una nueva versión para Atari de este difundido deporte. En esta versión, realizada en forma tridimensional, se ha tenido especial cuidado en la parte gráfica del juego, uno de los grandes déficits que poseía este juego anteriormente.

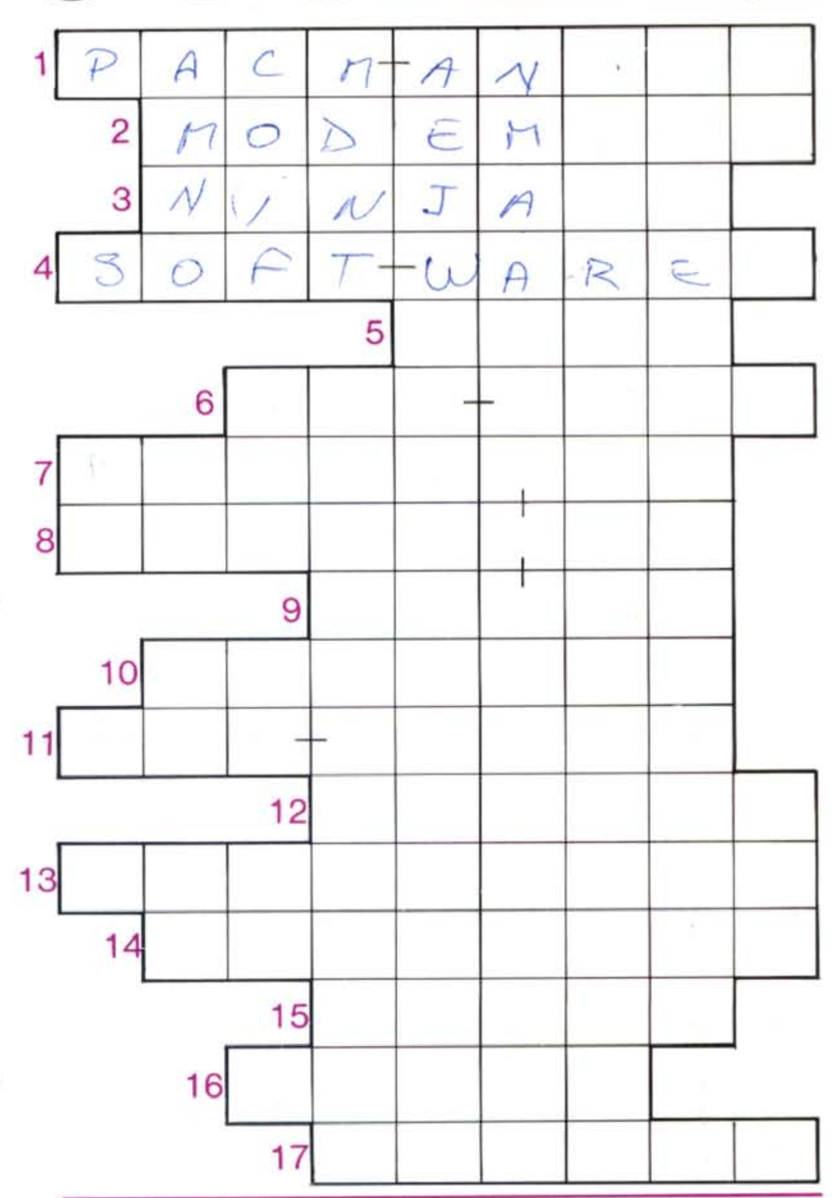
La forma de jugarlo es idéntica a sus presentaciones anteriores, pudiendo realizarse los mismos pases y ganbetas que en sus predecesores. Al comienzo puede escogerse la cantidad de jugadores que se enfrentarán, uno si es contra el computador o dos en caso contrario, y los países que cada uno de ellos defenderán. Una de las variantes más entretenidas que se incorpora al juego es la capacidad que tienen los arqueros de volar para atrapar la pelota.



SONIDO	6.1
GRAFICACION	6.6
ADICCION	6.4
PRESENTACION	6.6
PROMEDIO	6.425

TURBO PUZZLE:

Completa las palabras del esquema utilizando las sílabas que se dan, a razón de una letra por casilla. Una vez completo, podrás leer en la columna marcada, el nombre de un conocido juego en el que un valiente explorador busca tesoros escondidos.



A AS BLER BLUE BRU CE CION DEM DIS E FUN GRA HARD HAWK JA KET LA LE MA MAN MAX ME MO MO MIN PAC PRO RA!D RE RE RI RI RIA SA SEM SER SOFT TA TAR TE VE VER WA WA ZAN

1 Entretenido juego con un laberinto lleno de fantasmas.

2 Aparato que convierte los impulsos del ordenador en señales telefónicas.

3 Popular juego en el que hay que detener la invasión de los karatecas al templo del gran sacerdote.

4 Conjunto de programas y datos del computador.

5 Uno de los lenguajes del computador.

6 Salvemos los monos de la selva de los despiadados cazadores.

7 El más notable karateka, protagoniza este juego.

8 La parte del computador donde se almacenan todos los datos y programas.

9 El más popular de los computadores personales.

10 Disco para grabar y reproducir datos y programas.

11 Conjunto de mecanismos del computador.

12 Juego de combate aéreo tridimencional.

13 Instrucción de Basic para guardar un programa en diskette.

14 En este juego eres un hábil piloto de combate.

15 Calculo incluido en el computador que puede ser llamado por el programa en uso.

16 Conjunto de instrucciones ingresadas al computador.

17 Vamos a combatir al enemigo que domina el río en este juego.

BASIC 1

```
10 DIM NOMBRE$(800),TELEFONO(40),NOM$(
20),AUX$(50)
20 CANT=0:NOMBRE$(1)=" ":NOMBRE$(800)=
" ":NOMBRE$(2)=NOMBRE$
30 PRINT "↰          MENU PRINCIPAL"
40 PRINT "↓↓1)CARGAR TELEFONOS"
50 PRINT "↓↓2)CONSULTAR TELEFONO"
52 PRINT "↓↓3)GUARDAR ARCHIVO"
54 PRINT "↓↓4)LEER ARCHIVO"
56 PRINT "↓↓5)AGRANDAR ARCHIVO"
58 PRINT "↓↓6)IMPRIMIR ARCHIVO↓↓↓"
60 TRAP 60:PRINT "ELIJA OPCION";:INPUT
 OPCION
70 IF OPCION<1 OR OPCION>6 THEN 60
80 IF OPCION=1 THEN 100
82 IF OPCION=2 THEN 300
84 IF OPCION=3 THEN 500
86 IF OPCION=4 THEN 700
88 IF OPCION=5 THEN 900
90 GOTO 1100
100 CANT=CANT+1
110 PRINT "↰INGRESE NOMBRE: ";:INPUT N
OM$:NOMBRE$((CANT-1)*20+1,CANT*20)=NOM
$
120 TRAP 120:PRINT "↓INGRESE TELEFONO:
 ";:INPUT FONO:TELEFONO(CANT)=FONO
130 GOTO 30
300 PRINT "↰INGRESE NOMBRE A BUSCAR: "
;:INPUT NOM$:NOM$(LEN(NOM$)+1)="
              "
310 I=1
320 IF NOMBRE$((I-1)*20+1,I*20)=NOM$ T
HEN PRINT "↓↓EL NUMERO ES: ";TELEFONO(
I):GOTO 400
330 I=I+1:IF I>CANT THEN PRINT "EL NOM
BRE NO EXISTE":GOTO 400
350 GOTO 320
400 PRINT "↓↓OPRIMA RETURN PARA CONTIN
UAR";:INPUT NOM$:GOTO 30
500 PRINT "↰↓↓↓↓↓→→TRABAJANDO":OPEN #1
,8,0,"D:BORRENME"
510 FOR I=1 TO CANT
520 PRINT #1;NOMBRE$((I-1)*20+1,I*20);
TELEFONO(I):NEXT I:CLOSE #1
530 GOTO 30
700 PRINT "↰↓↓↓↓↓→→TRABAJANDO":OPEN #1
,4,0,"D:BORRENME":TRAP 740
710 CANT=0
720 INPUT #1,AUX$:CANT=CANT+1:NOMBRE$(
(CANT-1)*20+1,CANT*20)=AUX$(1,20)
725 TELEFONO(CANT)=VAL(AUX$(21))
730 GOTO 720
740 CLOSE #1:GOTO 30
900 PRINT "↰↓↓↓↓↓→→TRABAJANDO":OPEN #1
,9,0,"D:BORRENME"
910 FOR I=1 TO CANT
920 PRINT #1;NOMBRE$((I-1)*20+1,I*20);
TELEFONO(I):NEXT I:CLOSE #1
930 GOTO 30
1100 TRAP 1140:PRINT "↰↓↓↓↓↓→→TRABAJAN
DO":OPEN #1,9,0,"P:"
1110 FOR I=1 TO CANT
1120 PRINT #1,NOMBRE$((I-1)*20+1,I*20)
;TELEFONO(I):NEXT I:CLOSE #1
1130 GOTO 30
1140 ? "→↓↓↓   CONECTE LA IMPRESORA"
1150 GOTO 400
```

BASIC 2

```
500 PRINT "↰↓↓↓↓↓→→TRABAJANDO":OPEN #1
,8,0,"D:BORRENME"
510 FOR I=1 TO CANT
520 PRINT #1,NOMBRE$((I-1)*20+1,I*20):
PRINT #1,TELEFONO(I):NEXT I:CLOSE #1
530 GOTO 30
700 PRINT "↰↓↓↓↓↓→→TRABAJANDO":OPEN #1
,4,0,"D:BORRENME":TRAP 740
710 CANT=0
720 INPUT #1,NOM$:CANT=CANT+1:NOMBRE$(
(CANT-1)*20+1,CANT*20)=NOM$
725 INPUT #1,AUX:TELEFONO(CANT)=AUX
730 GOTO 720
740 CLOSE #1:GOTO 30
```



ASSEMBLER

```
0100 ;SAVE#D:TABLA.MAC
0110 ;
0120 ;
0130 ; RUTINA QUE ALMACENA
0140 ; ORDENADAMENTE LA LINEA
0150 ; INGRESADA, EN LA TABLA
0160 ; DE INSTRUCCIONES.
0170 ;
0180 ;
0190 PONGOTABLA
0200 ;
0210 ; DEFINO AL PUNTERO QUE VA A
0220 ; RECORRER LA TABLA DE
0230 ; INSTRUCCIONES
0240 ;
0250      JSR SETPOINTERAUX
0260 ;
0270 BUSCOINSTRUCCION
0280 ;
0290 ; RECORRO LA TABLA DE INSTRUC-
0300 ; CIONES PARA UBICAR LA NUEVA
0310 ; LINEA EN EL LUGAR QUE LE
0320 ; CORRESPONDE SEGUN SU NUMERO
0330 ; DE INSTRUCCION
0340 ;
0350      LDA POINTERAUX+1
0360      CMP FINTAB+1
0370      BNE DISTINTOS
0380      LDA POINTERAUX
0390      CMP FINTAB
0400      BNE DISTINTOS
0410      LDA CANTIDAD
0420      BNE COLOCO?
0430      JSR SETPOINTERAUX
0440      RTS
0450 ;
0460 ; SI LA NUEVA LINEA TIENE SU
0470 ; NUMERO DE INSTRUCCION MAYOR
0480 ; A LA DE LA ULTIMA ALMACENADA
0490 ; EN LA TABLA, LA UBICO AL FINAL
0500 ; DE LA TABLA SI ES QUE EXISTE
0510 ; ESPACIO EN ELLA.
0520 ;
0530 COLOCO?
0540      JSR HAYLUGAR
0550      LDA FINAUX+1
0560      CMP # >FINTABLA
0570      BCC LUGAROK
0580      LDA FINAUX
0590      CMP # <FINTABLA
0600      BCC LUGAROK
0610 ;
0620 ; COMO NO HUBO ESPACIO EN LA
0630 ; TABLA, IMPRIMO UN MENSAJE
0640 ; DE MEMORIA EXCEDIDA.
0650 ;
0660      JSR OVERFLOW
0670      JSR SETPOINTERAUX
0680      RTS
0690 LUGAROK
0700 ;
0710 ; ALMACENO LA INSTRUCCION EN LA
0720 ; TABLA OCUPANDO PARA ELLO LA
0730 ; RUTINA UBICO.
0740 ;
0750      JSR UBICO
0760 ;
0770 ; LE SUMO AL PUNTERO FINTAB UN 3
0780 ; QUE REPRESENTA LOS DOS BYTES
0790 ; DEL NUMERO DE INSTRUCCION MAS
0800 ; 1 BYTE DE LA CANTIDAD DE
0810 ; CARACTERES QUE TIENE LA LINEA
0820 ;
0830      CLC
0840      LDA FINTAB
0850      ADC #3
0860      STA FINTAB
0870      LDA FINTAB+1
0880      ADC #0
0890      STA FINTAB+1
0900 ;
0910 ; LE SUMO A FINTAB LA CANTIDAD
0920 ; DE BYTES QUE TIENE LA
0930 ; INSTRUCCION PARA QUE FINTAB
0940 ; APUNTE AL BYTE SIGUIENTE
0950 ; A LA ULTIMA LINEA.
0960 ;
0970      CLC
0980      LDA FINTAB
0990      ADC CANTIDAD
1000      STA FINTAB
1010      LDA FINTAB+1
1020      ADC #0
1030      STA FINTAB+1
1040      RTS
1050 ;
1060 ; COMPARO EL NUMERO DE LINEA
1070 ; CON LA INSTRUCCION DE LA TABLA
1080 ; PARA SABER SI DEBO O NO INTER-
1090 ; CALARLA EN ESTA POSICION
1100 ;
1110 DISTINTOS
1120      LDY #1
1130      LDA LINUM+1
1140      CMP (POINTERAUX),Y
1150      BEQ IGUAL1
1160      BCC MENOR1
1170 ;
1180 ; DEBO SEGUIR RECORRIENDO LA
1190 ; TABLA PUES LA NUEVA LINEA
1200 ; POSEE UN NUMERO DE INSTRUCCION
1210 ; MAYOR A LA DE LA TABLA
1220 ;
1230 MAYOR1
1240      LDY #2
1250      LDA (POINTERAUX),Y
1260      STA AUXSUM
1270      CLC
1280      LDA POINTERAUX
1290      ADC #3
1300      STA POINTERAUX
1310      LDA POINTERAUX+1
1320      ADC #0
1330      STA POINTERAUX+1
1340      CLC
1350      LDA POINTERAUX
```

```
1360       ADC AUXSUM
1370       STA POINTERAUX
1380       LDA POINTERAUX+1
1390       ADC #0
1400       STA POINTERAUX+1
1410       JMP BUSCOINSTRUCCION
1420 IGUAL1
1430       LDY #0
1440       LDA LINUM
1450       CMP (POINTERAUX),Y
1460       BEQ IGUAL2
1470       BCC MENOR1
1480       JMP MAYOR1
1490 MENOR1
1500       LDA CANTIDAD
1510       BNE EXPANDO?
1520       JSR SETPOINTERAUX
1530       RTS
1540 ;
1550 ; GENERO UN ESPACIO PARA
1560 ; REEMPLAZAR LA NUEVA LINEA
1570 ; EN LA TABLA, PUES TIENE MAS
1580 ; LONGITUD LA NUEVA QUE LA
1590 ; ANTERIOR
1600 ;
1610 EXPANDO?
1620       JSR HAYLUGAR
1630       LDA FINAUX+1
1640       CMP # >FINTABLA
1650       BCC LUGAROK1
1660       LDA FINAUX
1670       CMP # <FINTABLA
1680       BCC LUGAROK1
1690       JSR OVERFLOW
1700       JSR SETPOINTERAUX
1710       RTS
1720 LUGAROK1
1730       CLC
1740       LDA CANTIDAD
1750       ADC #3
1760       JSR EXTAB
1770       JSR UBICO
1780       RTS
1790 IGUAL2
1800       LDA CANTIDAD
1810       BNE NODELETE
1820       LDA POINTERAUX
1830       STA PUNCON
1840       LDA POINTERAUX+1
1850       STA PUNCON+1
1860       LDY #2
1870       CLC
1880       LDA (POINTERAUX),Y
1890       ADC #3
1900       JSR CONTAB
1910       JSR SETPOINTERAUX
1920       RTS
1930 NODELETE
1940       LDY #2
1950       LDA CANTIDAD
1960       CMP (POINTERAUX),Y
1970       BEQ IGUAL3
1980       BCC MENOR2
1990 MAYOR3
2000       LDA #0
2010       STA FINAUX
2020       STA FINAUX+1
2030       CLC
2040       LDA FINTAB
2050       ADC CANTIDAD
2060       STA FINAUX
2070       LDA FINTAB+1
2080       ADC #0
2090       STA FINAUX+1
2100       SEC
2110       LDA FINAUX
2120       SBC (POINTERAUX),Y
2130       STA FINAUX
2140       LDA FINAUX+1
2150       SBC #0
2160       STA FINAUX+1
2170       LDA FINAUX+1
2180       CMP # >FINTABLA
2190       BCC LUGAROK3
2200       LDA FINAUX
2210       CMP # <FINTABLA
2220       BCC LUGAROK3
2230       JSR OVERFLOW
2240       JSR SETPOINTERAUX
2250       RTS
2260 LUGAROK3
2270       SEC
2280       LDY #2
2290       LDA CANTIDAD
2300       SBC (POINTERAUX),Y
2310       JSR EXTAB
2320       JSR UBICO
2330       RTS
2340 IGUAL3
2350       JSR UBICO
2360       RTS
2370 MENOR2
2380       LDA #0
2390       STA PUNCON
2400       STA PUNCON+1
2410       LDY #2
2420       CLC
2430       LDA POINTERAUX
2440       ADC #3
2450       STA PUNCON
2460       LDA PUNCON+1
2470       ADC POINTERAUX+1
2480       STA PUNCON+1
2490       CLC
2500       LDA PUNCON
2510       ADC CANTIDAD
2520       STA PUNCON
2530       LDA PUNCON+1
2540       ADC #0
2550       STA PUNCON+1
2560       SEC
2570       LDA (POINTERAUX),Y
2580       SBC CANTIDAD
2590       JSR CONTAB
2600       JSR UBICO
2610       RTS
2620 HAYLUGAR
2630       CLC
2640       LDA FINTAB
2650       ADC CANTIDAD
2660       STA FINAUX
2670       LDA FINTAB+1
2680       ADC #0
2690       STA FINAUX+1
```

```
2700      CLC
2710      LDA FINAUX
2720      ADC #3
2730      STA FINAUX
2740      LDA FINAUX+1
2750      ADC #0
2760      STA FINAUX+1
2770      RTS
2780 ;
2790 ; ALMACENO LA INSTRUCCION EN
2800 ; LA TABLA DE INSTRUCCIONES
2810 ; CON EL FORMATO:
2820 ;
2830 ; BYTE 1:LO DEL NUMERO DE LINEA
2840 ; BYTE 2:HI DEL NUMERO DE LINEA
2850 ; BYTE 3:CANTIDAD DE CARACTERES
2860 ;        DE LA INSTRUCCION
2870 ; BYTE 4:PRIMER CARACTER DE LA
2880 ;        LINEA
2890 ; ......
2900 ; ......
2910 ; BYTE N:ULTIMO CARACTER DE LA
2920 ;        LINEA
2930 UBICO
2940      LDY #0
2950      LDA LINUM
2960      STA (POINTERAUX),Y
2970      INY
2980      LDA LINUM+1
2990      STA (POINTERAUX),Y
3000      INY
3010      LDA CANTIDAD
3020      STA (POINTERAUX),Y
3030      INY
3040      LDX #0
3050 L5
3060      LDA LINEA,X
3070      STA (POINTERAUX),Y
3080      INY
3090      INX
3100      CPX CANTIDAD
3110      BNE L5
3120      RTS
```

SOLUCION PUZZLE N° 1

(Edición N° 7)

A	T	A	R	I		D	A	T	A
R		S							R
C		T		L	O	G	O		K
H	E	R	O			O			A
O		O				T	H	E	N
N		N		Z		O			O
				O					I
D		S	A	R	G	O	N		D
O		O		R		S			
S	E	C	T	O	R		B	Y	E


Turbo
Software
LINEA LIGHT GUN PARA ATARI
Turbo Software
LINEA EDUCACIONAL PARA ATARI
Turbo Software
ATARI
La nueva generación de Software para Computadores Atari
ADQUIERALOS EN LOS SIGUIENTES PUNTOS DE VENTAS
• ANTOFAGASTA: COOPERCARAB / KW VIDEO / LA ESPAÑOLA • VIÑA DEL MAR: FALABELLA VIÑA / INSIS / MPR COMPUTACION • VALPARAISO: COMPUTRONIC • SANTIAGO: AUDIO BICICLETA INTERNAC / CASA ROYAL / CENTRO ATARI / COMERCIAL ESTADO / COMPUMANQUE / COMPUCENTER / FALABELLA AHUMADA / FALABELLA P. ARAUCO / IMACO / INFOGROUP / PC STORE / PETERSEN / ROLEC / SUPERMERCADOS UNIMARC / TASCO / VIDEO CLUB INTERNACIONAL • RANCAGUA: CASA ZUÑIGA • CURICO: MULTIHOGAR • TALCA: LIBRERIA "EL AHORRO" / MULTICENTRO / VIDEO CLUB CASSAL • CHILLAN: CASA EDISON • CONCEPCION: COOPERCARAB / DISMAR / DISMAR 2 / EQUS / PHANTER / RAPSODIA / SESCO • LOS ANGELES: DISTRIBUIDORA MERINO • ANGOL: SCORPIO • VICTORIA: CASA SIGMUND • TEMUCO: COMERCIAL MANQUEHUE / ESTABLECIMIENTOS GEJMAN / FALABELLA • PUCON: ELTIT • VILLARRICA: JOYERIA KETTERER • VALDIVIA: ELECTROMUSICA • LA UNION: IMPORTADORA COSMOS • OSORNO: CASA REAL / FOTO EXPRESS • PUERTO VARAS: ELECTRO HORN • PUERTO MONTT: COMERCIAL MANQUEHUE / DIMARSA • COYHAIQUE: FACI HOGAR • PUNTA ARENAS: BALFER LTDA.


JUEGOS 2
PARA COMPUTADORES ATARI
SOFTWARE
SOFTWARE
LA NUEVA GENERACION
TABLE FUTBOL - TWILIGHT WORLD
THE EXTIRPATOR - SUPER SOCCER
LITTLE TEWIL - TARZAN
AMAUROTE - CHIMERA
ES OTRO PRODUCTO M.P.M.
8 JUEGOS
ADQUIERELOS EN LOS SIGUIENTES PUNTOS DE VENTAS
• ANTOFAGASTA: COOPERCARAB KW VIDEO LA ESPAÑOLA • VIÑA DEL MAR: FALABELLA VIÑA INSIS MPR COMPUTACION • VALPARAISO: COMPUTRONIC • SANTIAGO: AUDIO BICICLETA INTERNAC CASA ROYAL CENTRO ATARI COMERCIAL ESTADO COMPUMANQUE COMPUCENTER FALABELLA AHUMADA FALABELLA P. ARAUCO IMACO INFOGROUP PC STORE PETERSEN ROLEC SUPERMERCADOS UNIMARC TASCO VIDEO CLUB INTERNACIONAL • RANCAGUA: CASA ZUNIGA • CURICO: MULTIHOGAR • TALCA: LIBRERIA "EL AHORRO" MULTICENTRO VIDEO CLUB CASSAL • CHILLAN: CASA EDISON • CONCEPCION: COOPERCARAB DISMAR DISMAR 2 EQUS PHANTER RAPSODIA SESCO • LOS ANGELES: DISTRIBUIDORA MERINO • ANGOL: SCORPIO • VICTORIA: CASA SIGMUND • TEMUCO: COMERCIAL MANQUEHUE ESTABLECIMIENTOS GEJMAN FALABELLA • PUCON: ELTIT • VILLARRICA: JOYERIA KETTERER • VALDIVIA: ELECTROMUSICA • LA UNION: IMPORTADORA COSMOS • OSORNO: CASA REAL FOTO EXPRESS • PUERTO VARAS: ELECTRO HORN • PUERTO MONTT: COMERCIAL MANQUEHUE DIMARSA • COYHAIQUE: FACI HOGAR • PUNTA ARENAS: BALFER LTDA
ATARI ES MARCA REGISTRADA DE ATARI CORPORATION